Southern University of Science and Technology
南方科技大学计算机科学与工程系
Department of Computer Science and Engineering

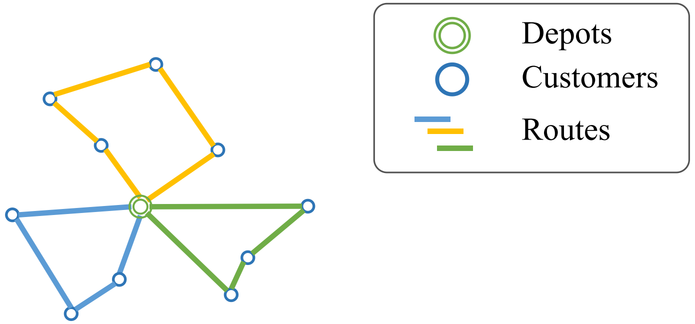
ICML
International Conference On Machine Learning

DPN: Decoupling Partition and Navigation for Neural Solvers of Min-max Vehicle Routing Problems

Authors:
Zhi Zheng, Shunyu Yao, Zhenkun Wang, Xialiang Tong, Mingxuan Yuan, Ke Tang
2024.7

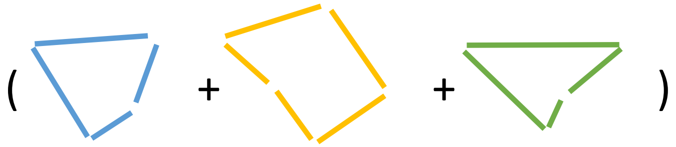
VRP & Min-max VRP

Vehicle Routing Problem (VRP) traverses all given customers and aims to minimize the **total route length**.

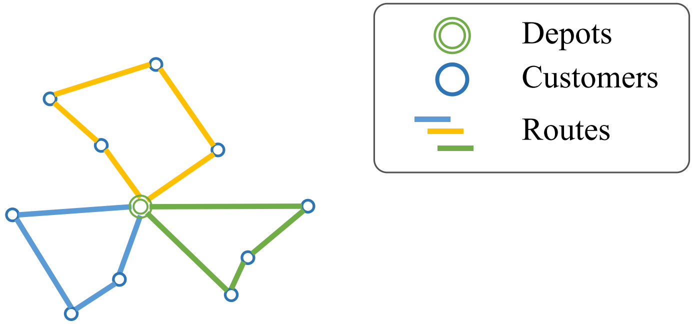


VRP:

minimize:



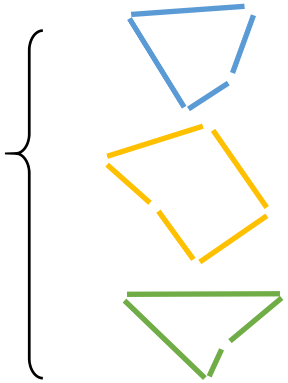
Min-max VRP restricts the number of routes and aims to minimize **the length of the longest route.**



Min-max VRP:

minimize:

Max



VRPs have a wide range of practical applications

Min-max VRPs are also of practical significance. These problems are hard to solve and lack exploration[1].

[1] Kumar, S. N. and Panneerselvam, R. A survey on the vehicle routing problem and its variants. Scientific research Publishing, 2012.

Min-max VRP

Defination of a min-max VRP instance

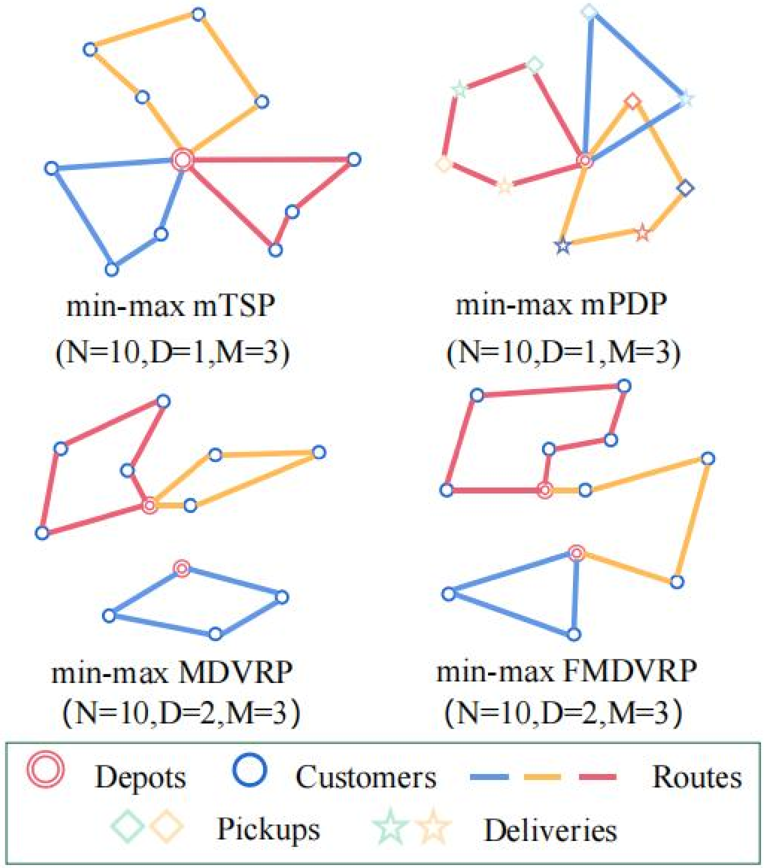
Min-max VRPs involved in this article

2.1. Sequential Planning for Solving Min-max VRPs

A min-max VRP instance with M routes (generated by M agents), D depots, and N customers is defined over a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Each $v_i \in \mathcal{V}$ represents a depot or a customer and $e_{ij} \in \mathcal{E}$ represents the edge between node v_i and v_j. The solution (i.e., a set of routes) $\mathcal{T}$ is formed by node indexes in $\mathcal{V}$, and each customer in $\mathcal{V}$ can only get visited once. Moreover, the number of routes in solution $\mathcal{T}$ is restricted to M (i.e., $\mathcal{T} = \{\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^M\}$). Each route $\boldsymbol{\tau}^i$ for $i \in \{1, \ldots, M\}$ only starts and ends at a depot. The objective function of the min-max VRP can be formulated as

$$\underset{\mathcal{T} \in \Omega}{\text{minimize}} \quad f(\mathcal{T}) = \max_{i \in \{1,\ldots,M\}} L(\boldsymbol{\tau}^i), \tag{1}$$

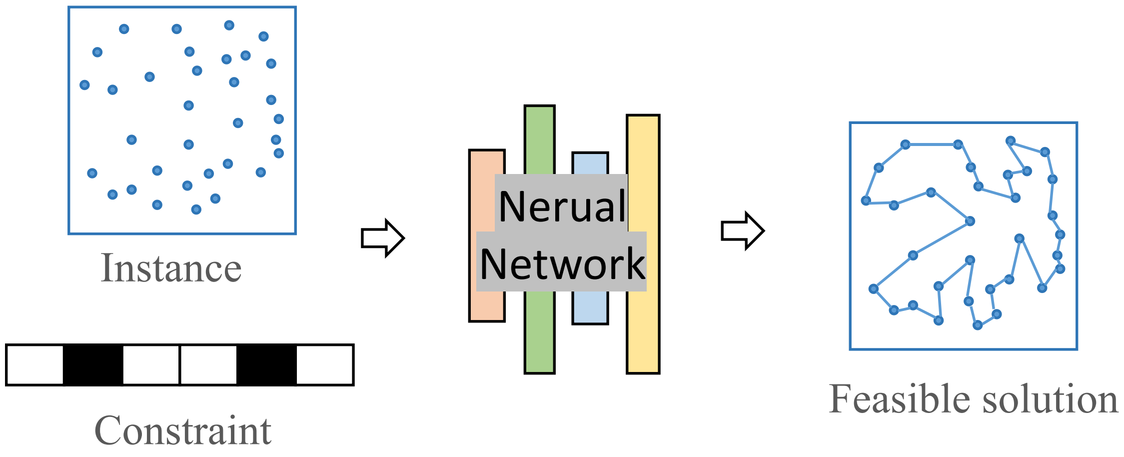
where Ω is a set consisting of all feasible solutions, and $L(\boldsymbol{\tau}^i)$ calculates the Euclidean length of route $\boldsymbol{\tau}^i$.



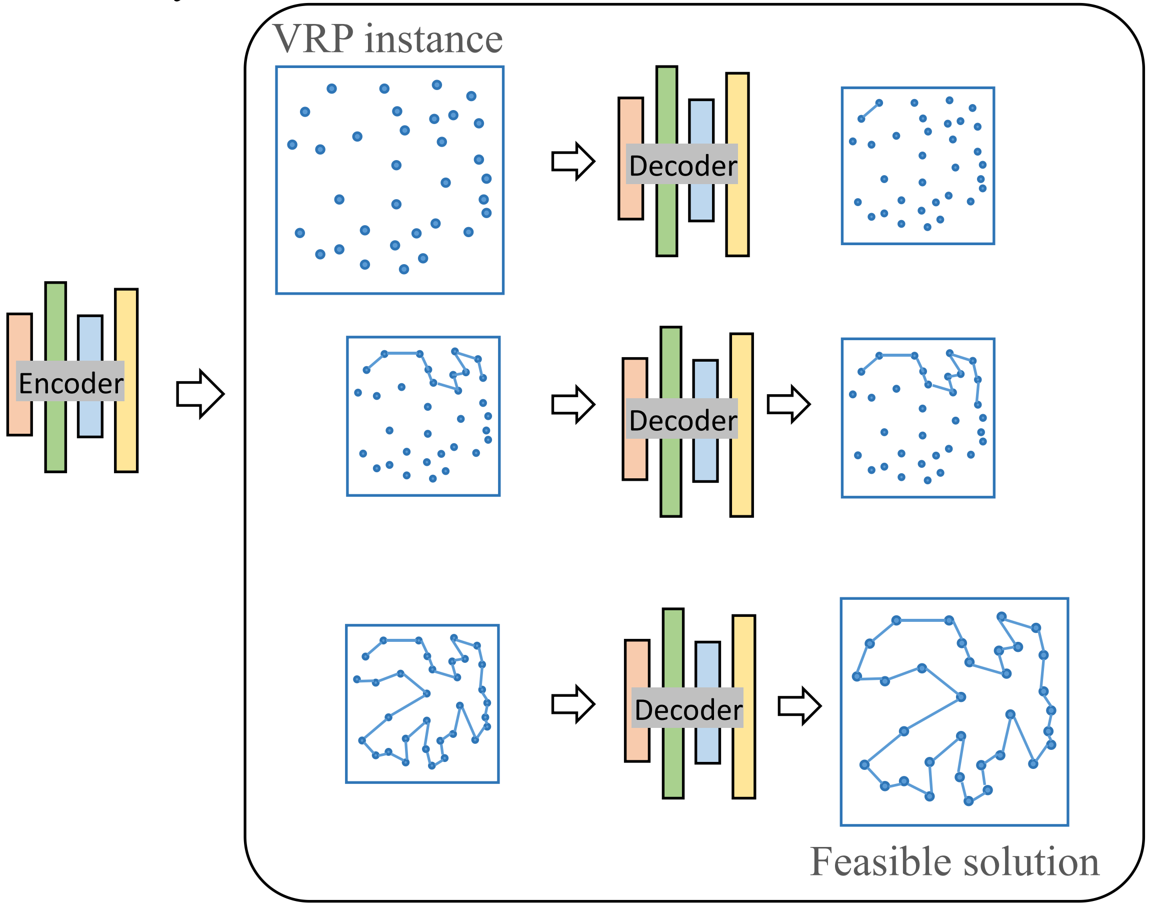
(a) Four examples of min-max VRPs

Neural solvers for VRP

VRPs are generally NP-hard, so solving them requires unacceptable time or special heuristics. Neural solvers for VRPs do not require expert experience and have fast solving speed, making it a widespread choice in recent years[1].



There are several categories of Neural solvers, including learning improvement heuristics, Neural Divide-and-conquer, etc. Among them, constructive solvers generate nodes in VRP solutions one by one and lead in both performance and efficiency.

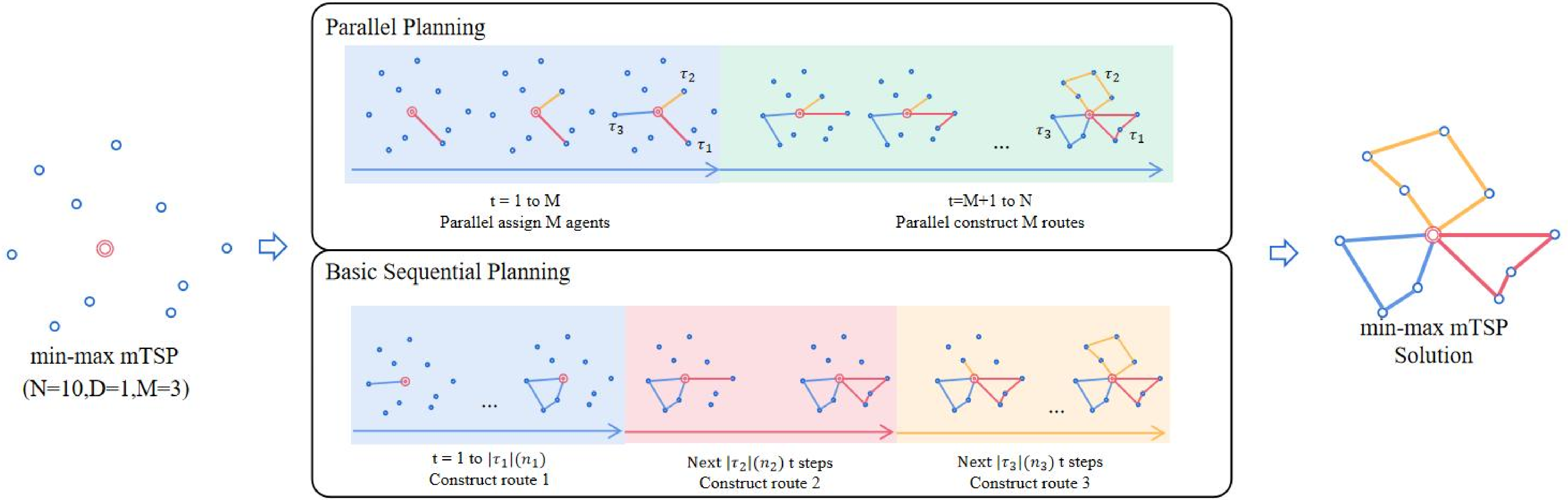


[1] Bengio, Yoshua, Andrea Lodi, and Antoine Prouvost. "Machine learning for combinatorial optimization: a methodological tour d'horizon." European Journal of Operational Research 290.2 (2021): 405-421.

Neural solvers for min-max VRP

Neural min-max VRP solvers include two-stage methods, learning improvement heuristics, parallel planning methods[1], and sequential planning methods[2].

Among them, sequential planning methods, sequentially construct the set of routes with a single model. This approach facilitates the exploration of the optimal solution.



[1] Cao, Y., Sun, Z., and Sartoretti, G. Dan: Decentral ized attention-based neural network to solve the min max multiple traveling salesman problem. arXiv preprint arXiv:2109.04205, 2021.
[2] Son, Jiwoo, et al. "Equity-Transformer: Solving NP-Hard Min-Max Routing Problems as Sequential Generation with Equity Context." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 38. No. 18. 2024.

Motivation - DPN

Motivation: Existing sequential planning methods are **without specific design** in both model structure and training scheme, resulting in deficiencies in the representation ability of embeddings.

The tasks of *assigning customers to M routes* (i.e., partition) and *optimizing the routing of customers assigned to each route* (i.e., navigation) are considered simultaneously in solving min-max VRPs.[1]

Partition. Customers and depots assigned to the route $\boldsymbol{\tau}^i$ for $i \in \{1, \ldots, M\}$ forms a partition of $\mathcal{G}$. Each sub-graph is denoted as $\mathcal{G}^i$. The partition function $P_{\theta,M}(\mathcal{G}) = \{\mathcal{G}^1, \ldots, \mathcal{G}^M\}$ with parameter θ generates sub-graph partitions with

$$\bigcap_{i\in\{1,\ldots,M\}} \mathcal{G}^i \subseteq \text{Depots}, \quad \bigcup_{i\in\{1,\ldots,M\}} \mathcal{G}^i = \mathcal{G}. \quad (3)$$

Navigation. The navigation task (generally being TSP) optimizes routings in each sub-graph $\mathcal{G}^i$ for $i \in \{1, \ldots, M\}$. For sequential planning methods, if the number of nodes in $\mathcal{G}^i$ is n^i, the navigation policy π_θ of $\mathcal{G}^i$ can written as follows:

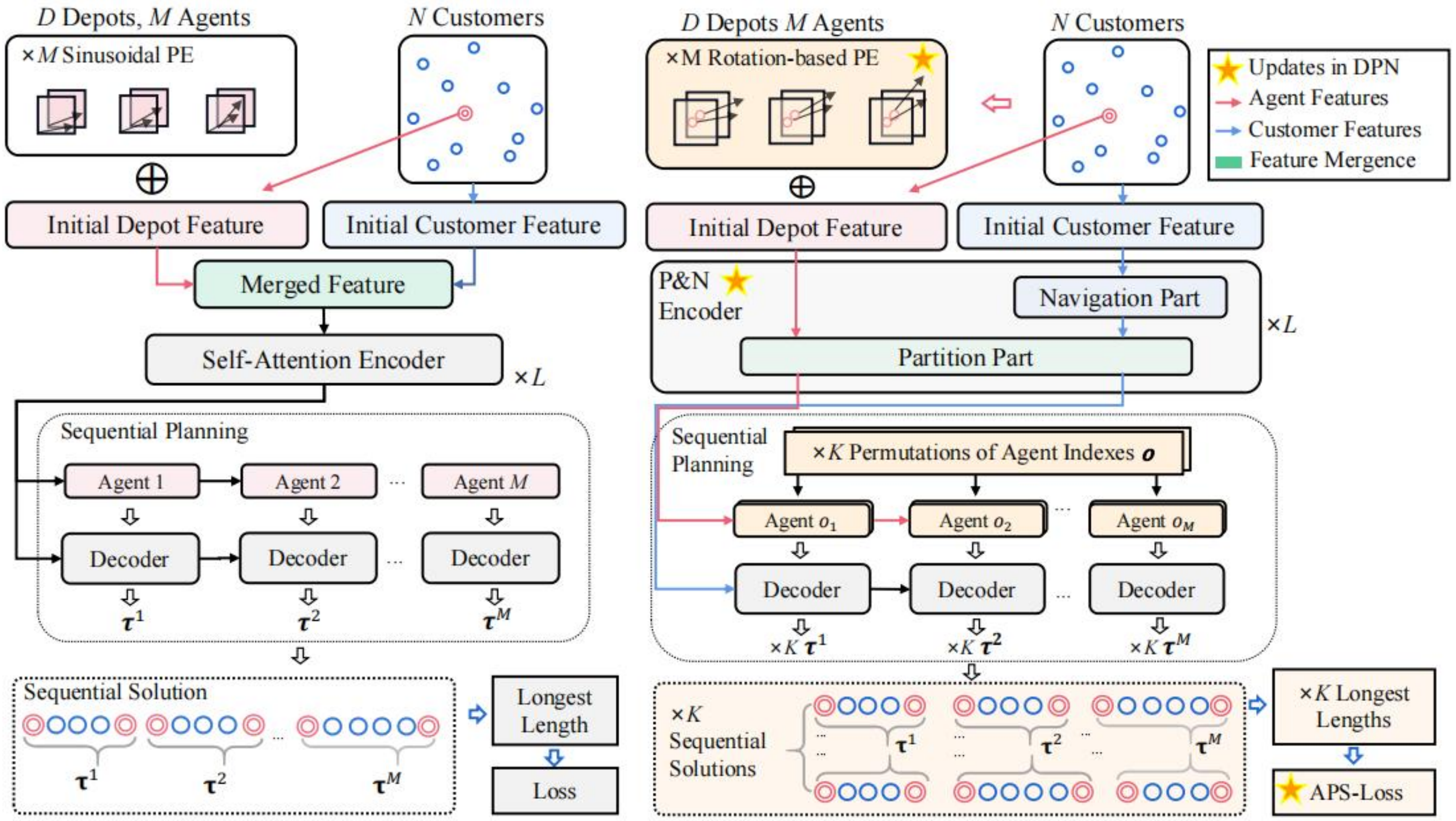
$$\pi_\theta(\boldsymbol{\tau}^i|\mathcal{G}^i) = \prod_{t=1}^{n^i} p(\boldsymbol{\tau}^i(t)|\boldsymbol{\tau}^i(1:t), \mathcal{G}^i, \theta), \quad (4)$$

In single-stage sequential planning processes, representations of partition and navigation tasks are processed in agent embeddings, depot embeddings, and customer embeddings.

Partition and navigation have different requirements, so decoupling their representations will improve the representation ability.

[1] Vandermeulen, I., Roderich, G., and Andreas, K. Balanced task allocation by partitioning the multiple traveling salesperson problem. In 2019 ICAAMS, pp. 1479– 1487. ACM, 2019.

Methodology - DPN



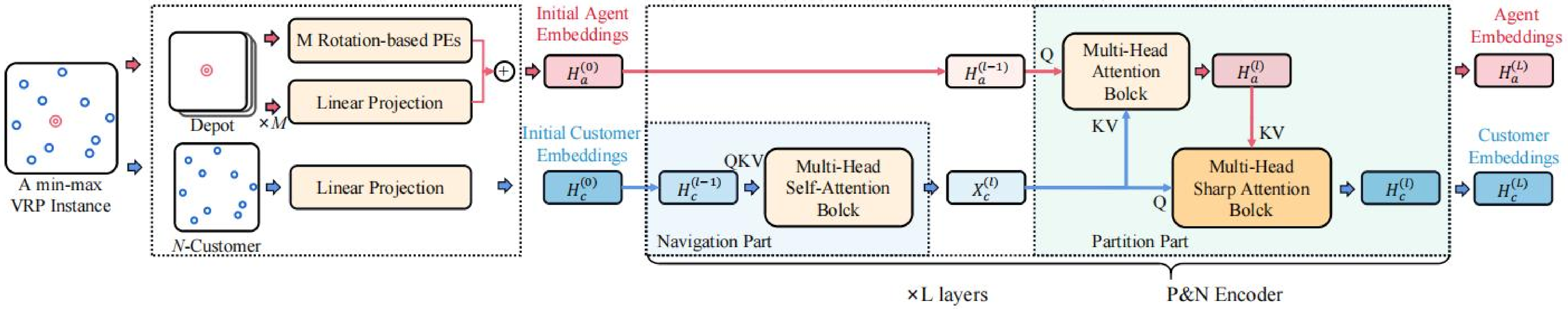
(b) Existing framework (c) DPN (Ours)

This paper aims to fully exploit the problem-specific **properties of min-max VRPs**, particularly the requirements of decoupling **partition** and **navigation.**

DPN proposes a novel attention-based P&N Encoder, an APS-Loss to facilitate training, and a Rotation-based PE for representation ability.

Methodology - DPN - P&N Encoder

To decouple the representation of partition and navigation tasks, P&N Encoder designs separate parts to process different embeddings. In min-max mTSP, each P&N Encoder layer consists of a navigation part and a partition part.



In order to maintain consistency with the characteristics of the partition process, we also utilize the multi-head *sharp attention* in the partition part as follows:

$$\mathrm{MHSA}(X, C) = \mathrm{Concat}(S_1, \ldots, S_H) W_P,$$
$$\text{where} \quad S_i = \mathrm{Softmax}\left(X W_Q (C W_K)^\intercal\right) C W_V.$$

Methodology - DPN - APS-Loss

The P&N Encoder explores problem-specific model structures to fit the requirements of min-max VRP. Moreover, it is also important to explore problem-specific loss functions to facilitate training.

The proposed DPN presents an agent-permutation-symmetry (APS) trait in solving min-max VRPs, which means changing the construction order of the M agents without changing the optimal min-max VRP solution.

Using APS, the APS-Loss proposed in this article obtains K Monte Carlo samples to represent the route construction orders, and then generates K solutions. The average objective function of these K solutions is used as the baseline for reinforcement learning. The formula of APS-Loss is as follows:

APS-equipped baseline

$$b(\mathcal{G}) = \frac{1}{K}\sum_{k=1}^{K} \max_{i\in\{1,\dots,M\}} L(\boldsymbol{\tau}^i \sim p_\theta(\cdot|\mathcal{G}^{o_i^{(k)}})). \tag{15}$$

APS-Loss

$$\pi'_\theta(\mathcal{T}|\mathcal{G}, M, \boldsymbol{o}^{(k)}) = \prod_{i=1}^{M} \pi_\theta(\boldsymbol{\tau}^i|\mathcal{G}^{o_i^{(k)}}), \tag{16}$$

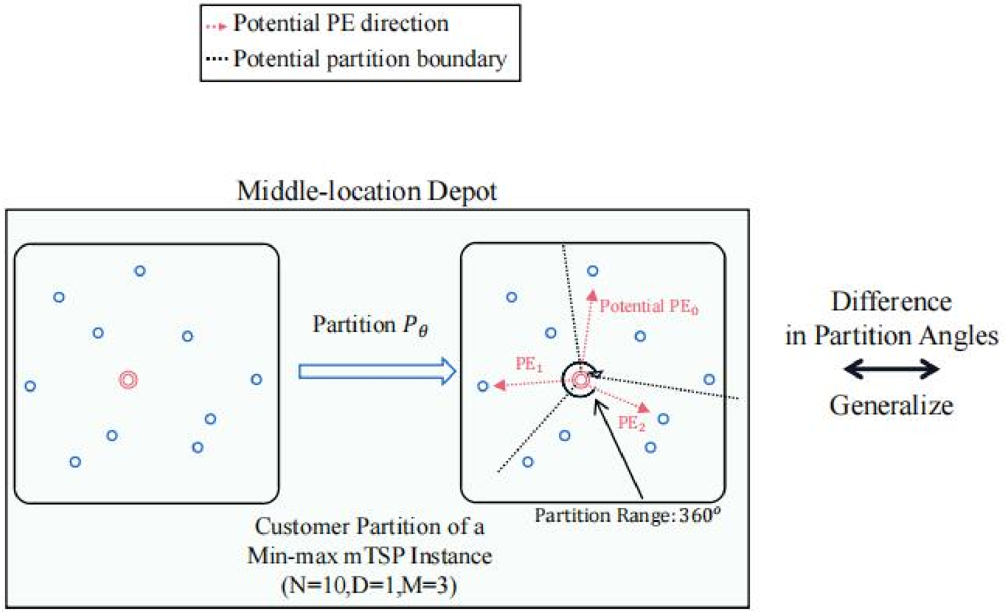
$$\nabla_\theta \mathcal{L}(\mathcal{G}) = -\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\pi'_\theta(\mathcal{T}|\mathcal{G},M,\boldsymbol{o}^{(k)})} \left[(f(\mathcal{T}) - b(\mathcal{G}))\nabla_\theta \log \pi'_\theta(\mathcal{T}|\mathcal{G}, M, \boldsymbol{o}^{(k)})\right], \tag{17}$$

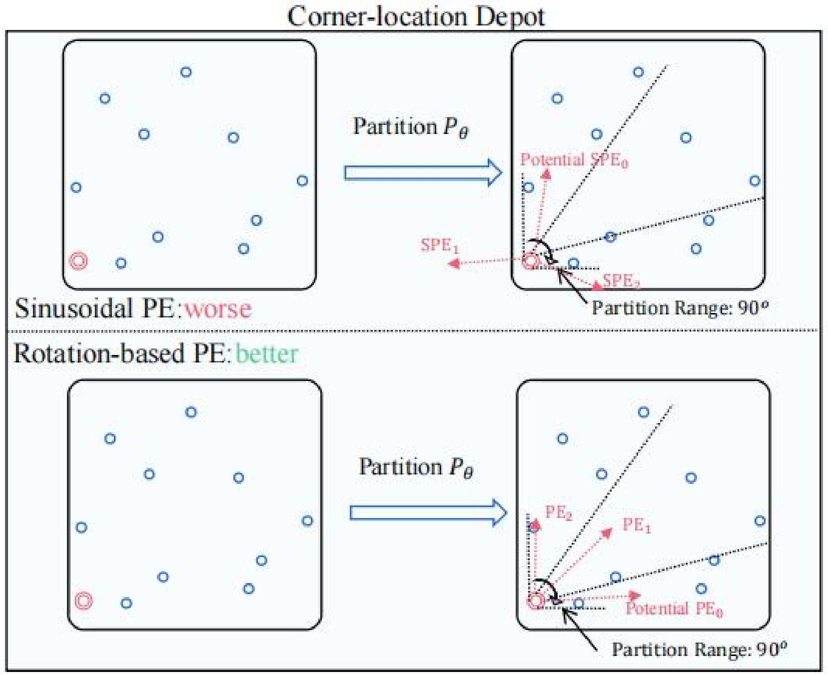
Methodology - DPN - Rotation-based PE

In addition, this article also introduces rotation-based positional encodings to adapt to the requirements of partition tasks. The formula of Rotation-based PE is modified from the existing sinusoidal PE (SPE) as follows[1]:

$$\mathrm{SPE}(m,q)=\begin{cases}\sin(m/10,000^{\frac{\lfloor q/2\rfloor}{d}}), q\equiv 0(mod\ 2)\\ \cos(m/10,000^{\frac{\lfloor q/2\rfloor}{d}}), q\equiv 1(mod\ 2)\end{cases}. \quad \Rightarrow \quad \mathrm{PE}(m,q)=\mathrm{Re}\left[(\boldsymbol{x}_d W_a+b_a)\exp(im/1,000^{\frac{\lfloor q/2\rfloor}{d}})\right],$$

Rotation-based PE can meet the requirement of modeling different partition strategies for min-max VRP instances with different depot locations.





[1] Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063, 2024.

Results

Min-max mTSP100(N=99,D=1)									
M=	5			7			10		
Methods	Obj.	Gap	Time	Obj.	Gap	Time	Obj.	Gap	Time
HGA	**2.1893**	-	20m	1.9963	0.1240%	16m	1.9507	0.0273%	14m
LKH3	2.1924	0.1410%	16m	**1.9939**	-	17m	**1.9502**	-	17m
OR-Tools(600s)	2.3477	7.2346%	5m	2.1627	8.4671%	6m	2.1465	10.068%	7m
DAN	2.6995	23.305%	40s	2.3115	15.930%	42s	2.1556	10.534%	46s
Equity-Transformer	2.3042	5.2456%	<1s	2.0487	2.7480%	<1s	1.9583	0.4153%	<1s
Equity-Transformer-×8aug	2.2563	3.0577%	<1s	2.0225	1.4345%	<1s	1.9534	0.1652%	<1s
DPN	2.2704	3.7017%	<1s	2.0335	1.9860%	<1s	1.9587	0.4377%	<1s
DPN-×8aug	2.2346	2.0703%	<1s	2.0143	1.0223%	<1s	1.9534	0.1671%	<1s
DPN-×8aug-×16per	2.2314	1.9240%	1s	2.0126	0.9388%	1s	1.9532	0.1542%	1s

Min-max mPDP100(N=100,D=1)									
M=	5			7			10		
Methods	Obj.	Gap	Time	Obj.	Gap	Time	Obj.	Gap	Time
OR-Tools(600s)	14.315	309.48%	4h	14.486	378.96%	5h	14.500	438.07%	5h
Equity-Transformer	5.5571	58.958%	<1s	4.4831	48.234%	<1s	3.7483	39.092%	<1s
Equity-Transformer-×8aug	5.0735	45.123%	1s	4.1152	36.069%	1s	3.4411	27.690%	1s
DPN	3.6500	4.4054%	<1s	3.1404	3.8364%	<1s	2.7998	3.8933%	<1s
DPN-×8aug	3.5123	0.4673%	1s	3.0430	0.6165%	1s	2.7101	0.5647%	1s
DPN-×8aug-×16per	**3.4960**	-	2s	**3.0244**	-	2s	**2.6949**	-	2s

Results

	Min-max mTSP200			Min-max mTSP500			Min-max mTSP1,000		
M=	10	15	20	30	40	50	50	75	100
Methods	Obj.	Obj.	Obj.	Obj.	Obj.	Obj.	Obj.	Obj.	Obj.
HGA	1.9861	**1.9628**	**1.9627**	**2.0061**	2.0061	2.0061	**2.0448**	2.0448	2.0448
LKH3	**1.9817**	1.9628	1.9628	2.0061	2.0061	2.0061	2.0448	2.0448	2.0448
OR-Tools(600s)	2.3711	2.3687	2.3687	8.9338	8.9356	8.9308	16.436	16.436	16.436
NCE*	2.07	1.97	1.96	2.07	2.01	2.01	2.13	2.07	2.05
DAN	2.3586	2.1732	2.1151	2.2345	2.1610	2.1465	2.3390	2.2544	2.2394
ScheduleNet*	2.35	2.13	2.07	2.16	2.12	2.09	2.26	2.17	2.16
Equity-Transformer-F-×8aug	2.0500	1.9688	1.9631	2.0165	2.0084	2.0068	2.0634	2.0531	2.0488
DPN-F-×8aug	2.0030	1.9647	1.9628	2.0065	2.0061	2.0061	2.0452	2.0448	2.0448
DPN-F-×8aug-×16per	1.9993	1.9640	1.9628	2.0061	**2.0061**	**2.0061**	2.0450	**2.0448**	**2.0448**
	Min-max mPDP200			Min-max mPDP500			Min-max mPDP1,000		
M=	10	15	20	30	40	50	50	75	100
Methods	Obj.	Obj.	Obj.	Obj.	Obj.	Obj.	Obj.	Obj.	Obj.
OR-Tools(600s)	45.299	45.387	45.131	140.85	140.92	140.79	280.22	280.19	280.14
Equity-Transformer-F-×8aug	4.9143	3.8186	3.3417	4.4619	3.7723	3.4455	4.9328	3.9198	3.5241
Equity-Transformer-F-sample*	4.68	3.65	3.18	4.11	3.52	3.23	4.73	3.77	3.38
DPN-F-×8aug	3.3227	2.8630	2.6735	3.1615	3.0264	2.9379	3.2802	3.0673	3.0000
DPN-F-×8aug-×16per	**3.2959**	**2.8363**	**2.6519**	**3.0878**	**2.9510**	**2.8690**	**3.2263**	**2.9811**	**2.9114**

Results

	MDVRP50(N=50), D=6						MDVRP100(N=100), D=8					
M=	3		5		7		5		7		10	
Methods	Obj.	Time	Obj.	Time	Obj.	Time	Obj.	Time	Obj.	Time	Obj.	Time
CE*	2.25	40m	1.53	17m	1.28	11m	1.85	50m	1.43	1h	1.18	1h
OR-Tools*	2.64	4m	1.68	5m	1.36	5m	2.17	6h	1.60	3h	1.29	2h
NCE*	2.25	3m	1.53	4m	1.28	5m	1.86	19m	1.43	20m	1.18	26m
DPN-×8aug-×16per	2.1491	<1s	1.4431	<1s	1.2012	<1s	1.8056	1s	1.4099	1s	1.1527	1s
DPN-F-×8aug-×16per	**2.1404**	1s	**1.4394**	1s	**1.1969**	1s	**1.7936**	2s	**1.4001**	2s	**1.1429**	2s

	FMDVRP50(N=50), D=6						FMDVRP100(N=100), D=8					
M=	3		5		7		5		7		10	
Methods	Obj.	Time	Obj.	Time	Obj.	Time	Obj.	Time	Obj.	Time	Obj.	Time
CE*	2.07	35m	1.41	15m	1.19	9m	**1.74**	6h	**1.34**	4h	**1.09**	1h
OR-Tools*	2.39	4m	1.56	4m	1.27	4m	2.00	51m	1.51	54m	1.20	57m
NCE*	2.08	2m	1.40	3m	1.19	4m	1.75	11m	**1.34**	16m	**1.09**	22m
ScheduleNet*	2.61	9m	1.86	9m	1.57	10m	2.32	1h	1.86	1h	1.54	1h
DPN-×8aug-×16per	2.0471	<1s	1.3869	<1s	**1.1619**	<1s	1.7694	1s	1.3708	1s	1.1028	1s
DPN-F-×8aug-×16per	**2.0429**	**1s**	**1.3856**	1s	1.1649	1s	1.7638	2s	1.3642	2s	1.1012	2s

DPN demonstrates advantages in min-max mTSP, min-max mPDP, min-max MDVRP, and min-max FMDVRP with at most 1000 nodes.

The original article also provides ablation studies and experiments on benchmark datasets, very-large-scale instances, and instances with different distributions.

Thanks

for your attention ~